Docket No. 1152-002U **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of	:	Customer Number: 29973
Stephen M. HITCHEN.	:	Confirmation Number: 8250
Application No.: 09/992,582	:	Group Art Unit: 2167
Filed: November 16, 2001	:	Examiner: Luke S. Wassum

For: COLLABORATIVE FILE ACCESS MANAGEMENT SYSTEM

REPLY BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Reply Brief is submitted under 37 C.F.R. § 41.41 in response to the EXAMINER'S ANSWER dated May 9, 2007.

The Examiner's response to Appellant's arguments submitted in the Appeal Brief of January 16, 2007, raises additional issues and underscores the factual and legal shortcomings in the Examiner's rejections. In response, Appellant relies upon the arguments presented in the Appeal Brief of January 16, 2007, and the arguments set forth below.

The Applicants previously had shown with clarity that nowhere in paragraphs [0140] and [0141] of Graham is it stated or suggested that the "filter-driver" of Graham can "suppress" a file system request as required by the plain language of the Applicants' independent claims. In support, the Applicants referred the Examiner to the first sentence of paragraph [0141] of

Graham in which it is stated, "*This filter driver allows the client module 230 to intercept and modify file requests to and from file servers as required*" in order to highlight the distinction between mere modification of a file I/O request and a complete quashing of the request as expressly claimed in each of Applicants' independent claims. Still, on page 14 of the Examiner's Answer, the Examiner sets forth an argument that the "suppressing the file I/O request" is shown not alone by paragraph [0140] and [0141] of Graham as set forth in page 5 of the Final Office Action dated April 4, 2006, but now only when placed in context by paragraphs [0118] and [0119] of Graham.

The Examiner, however, does not produce an accurate and clear picture of the Graham reference by selectively pulling portions of the referenced paragraphs without context. Placed in context, a different conclusion will be drawn. For the convenience of the Honorable Board, the entirety of paragraphs [0116] through [0119] of Graham are reproduced below. The Applicants have chosen to include paragraph [0116] as it demonstrates the heading "Client Module 230" which is defined explicitly within Graham to be separate and distinct from the filter-driver of paragraph [0141] even though the Examiner has cited paragraph [0119] in order to explain the functionality of the filter driver of paragraph [0141].

> [0116] B. Client Module 230
>
> [0117] In the preferred embodiment, the client module 230 evaluates the usage policy inside the kernel of the client's OS. These usage rights can include all aspects of a user's interaction with a file, including, but not limited to: copy/cut/paste, printing, screen capture, launch application control and auditing. Since policy is enforced within the kernel of the OS, malicious users are prevented from compromising the usage policy at runtime because direct access to the kernel is not possible without crashing the system.
>
> [0118] The usage rights are enforced through the trapping of kernel-level OS calls that are tied to a process list. This trapping is accomplished through an understanding of the APIs and other system-level calls that are supplied by an operating system to an application. These APIs and system calls allow for an application to run correctly under an operating system and to take advantage of the functionality that the operating system has to offer. The client module 230 is between the application and operating system, which allows the client module 230 to understand

> what the application requests from the OS, and modify to these requests an needed to control how information is used.
>
> [0119] When a call is made to the client's OS that has been identified as potential source of data movement, the client module 230 intercepts the call between the application and the OS. As the interception occurs, a list of tagged processes (open files, visible windows, executing applications) is checked to see if the system call will result in protected information being acted upon. If it is determined that the request's source was not acting on a process relating to the applicable usage policies, then the call is allowed to proceed without any further action by the client module 230. However, if it is determined that the call will result in a protected file being acted on, then the usage rights of that particular file are evaluated from within the kernel. If the call is within the allowed functionality set forth in the usage policy, then it is allowed. If the call is not allowed, then the call is blocked and the user is notified.

In reference to paragraph [0119], it is shown that an operating system "call" is intercepted by the client module 230 "between the application and the OS". The filter-driver of paragraph [0141], however, is an operating system level structure and therefore different from the structure performing the "interception". Additionally, while paragraph [0119] refers to the blocking of a "call" when the "call is not allowed", no further action is performed. In a vacuum, then, paragraph [0119] stands for nothing more than the blocking of a function call in an operating system issued by an unauthorized user not permitted by a "usage policy". More importantly, the Examiner has glued together one teaching directed to a filter-driver in paragraph [0141] with another teaching directed to a client module and not a filter-driver in paragraph [0119]. Yet, the Examiner somehow believes that the two are synonymous in stating in the last line of page 14 of the Examiner's Answer that "the filter-driver...includes the claimed 'suppressing feature'".

There is no suggestion anywhere in the art as provided by the Examiner that weaves together the "suppressing" of an intercepted "file I/O request" followed by the automatic extraction of digital rights management data appended to the file, the provision of the requested filed to the authoring application, and the management of access to the file based upon the extracted digital rights data. Rather, the Examiner merely has found bits and pieces of words in

a document in the absence of context and combined those bits and pieces without sound reason to produce a dissimilar shadow of the Applicants' claimed invention.

In page 16 of the Examiner's Answer, the Examiner further relies upon paragraphs [0021] and [0022] of Graham to provide an explanation of the operation of the filter driver of paragraph [0140] and [0141]. Specifically, in page 16 the Examiner sets forth the argument that the proxy system of paragraphs [0021] and [0022] includes the filter driver and therefore text ascribed to the "proxy system" in paragraphs [0021] and [0022] can be placed squarely within the domain of the operation of the filter-driver. A mere glance at Figure 2 and a companion reading of paragraph [0069], however, will make clear that the proxy system 110 referred to by the Examiner is part of the "server domain" separated from the "client enforcement domain" (including the client module 230) by a complete computer communications network!

Paragraph [0064] is reproduced as follows:

> [0069] A proxy system 110 in accordance with the present invention includes a set of servers running server-side proxy file management functionality that applies flexible authorization and access control policies over managed content, such as files stored in a content source 160. The server-side proxy file management functionality may take the form of a content subsystem or program, described in more detail with respect to FIG. 2 and FIG. 3. Unlike current security services, policies in accordance with the present invention not only allow administrators to map users to allowable access, but to base access on run-time environmental conditions. Users can be flexibly organized within the system, wherein the policies associated with a user may vary from file to file. The policy infrastructure interfaces with widely deployed network services, so provides for easy integration into existing networked file systems.

Thus, the proxy system 110 in no uncertain terms excludes the client module 230 as discussed in paragraph [0074] as being part of the separate "Client 150". Specifically, paragraph [0074] reproduced in its entirety states:

> [0074] A client 150 includes a client device and a client module. A client devices in accordance with the present invention may be any of a variety of types of devices, including personal computer, workstation, server, personal digital assistant (PDA), telephone (including, cellular telephone), pagers, Web enabled appliances, or other network enabled devices. The client module 230 (see FIG. 2) acts on behalf of the user in obtaining credentials and managing security-related material from proxy system 110. This information is used over the course of a session to gain

> access protected content from content source 160, and to protect the content from exposure to adversaries on the network. In addition to protecting data from unauthorized users, the client module hosted on client 150 enforces use policies. Use policies limit the kinds of Is operations allowed on protected content. For example, a particular user may not be permitted to print an accessed file. Use policies are communicated to the client 150 at access time, and enforced over the lifetime of the user's session.

Of course, the illustration of Figure 2 and the companion text of paragraph [0069] clearly demonstrate that the filter driver of paragraph [0141] and the proxy system 110 of paragraph [0021] are not related in the slightest way.

Finally, the Examiner's entire rebuttal of Applicants' "motivation to combine" argument beginning on page 17 rests on the Examiner's faulty equating of the client module 230, the proxy system 110 and the filter driver as a singular entity when in fact, each are separate and in the case of the proxy system 110 and the client module 230--completely unrelated and separated by an entire computer communications network. As such, the Examiner has yet to reasonably show how one of ordinary skill at the time of the Applicants' invention would take the teaching of blocking an unauthorized operating system call shown by the client module 230 with the extraction of digital rights management data from a designated file associated with the operating system call (not shown by any disclosure in Graham), while still providing the designated file to a requestor and concurrently managing subsequent access to the file according to the extracted digital rights management data. So much has been expressly claimed by the Applicants, however.

The Examiner still has never directly addressed that the filter driver so heavily relied upon the Examiner in paragraph [0141] explicitly teaches away from the Applicants' invention

by merely modifying and forwarding a file I/O request, whilst the Applicants' claims require the complete suppression of the file I/O request.

For the reasons set forth in the Appeal Brief of January 16, 2007, and for those set forth herein, Appellant respectfully solicits the Honorable Board to reverse the Examiner's rejections under 35 U.S.C. § 103. To the extent necessary, a petition for an extension of time under 37 C.F.R. § 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 503839, and please credit any excess fees to such deposit account.

Date: July 9, 2007

Respectfully submitted,

/Steven M. Greenberg/

Steven M. Greenberg
Registration No. 44,725
Carey, Rodriguez, Greenberg & Paul, LLP
950 Peninsula Corporate Circle, Suite 3020
Boca Raton, FL 33487
Tel: (561) 922-3845
Facsimile: (561) 244-1062
CUSTOMER NUMBER 29973